

Data Envelopment Analysis in Stata

Yong-bae Ji
Korea National Defense University
Seoul/Republic of Korea
jyb77@hanmail.net

Choonjoo Lee
Korea National Defense University
Seoul/Republic of Korea
sarang90@kndu.ac.kr

Abstract. This paper introduces a user written Data Envelopment Analysis(DEA) program in Stata. DEA is a linear programming method for assessing the efficiency and productivity of units called Decision Making Units(DMUs) in DEA. Over the last decades DEA has gained considerable attention as a managerial tool for measuring performance of organizations and it has been used widely for assessing the efficiency of public and private sectors such as banks, airlines, hospitals, universities, defense firms, and manufactures. The DEA program in Stata will allow users to conduct the standard optimization procedure but also more extended managerial analysis. The dea program developed in this paper selects the chosen variables from a Stata data file and constructs a linear programming model based on the selected DEA options. Examples are given to illustrate how one could use the code to measure the efficiency of DMU.

Keywords: st0001, Stata, Data Envelopment Analysis(DEA), Linear Programming, non-parametric, efficiency, Decision Making Units(DMUs)

1 Introduction

This paper introduces a new application in Stata for performance measurement of Decision Making Units (DMUs) using Data Envelopment Analysis (DEA) techniques. DEA is a non-parametric linear programming method for assessing the efficiency and productivity of DMUs. DEA application areas have grown since it was first introduced as managerial and performance measurement tool in the late 1970s. Since then, new applications with more variables and complicated models are being introduced.

Stata equipped with DEA will provide the user with a new nonparametric tool to analyze productivity data. From within Stata, users will be able to produce DEA scores and analyze them. Since the second-stage DEA analysis and DEA efficiency estimates involves the statistical inference, DEA users want to have a software package that can analyze the whole processes in the same system instead of juggling between a DEA program and a statistical software that uses the DEA scores as dependent variable to find the influential variables in the second-stage DEA analysis.

The main purpose of this paper is to implement a dea program in Stata. The paper unfolds as follows. The next section describes the DEA models and calculations in DEA. The remainder of this paper illustrates the features and options of the `dea` program.

2 The Basics of Data Envelopment Analysis

DEA is a method for measuring efficiency of DMUs using linear programming techniques to envelop observed input-output vectors as tightly as possible (Boussofiane, Dyson, and Thanassoulis 1991). DEA allows multiple inputs-outputs to be considered at the same time without any assumption on data distribution. In each case, efficiency is measured in terms of a proportional change in inputs or outputs. DEA model can be subdivided into input-oriented model which minimizes inputs while satisfying at least the given output levels and output-oriented model which maximizes outputs without requiring more of any of the observed input values.

DEA models can also be subdivided in term of returns to scale by adding weight constraints. Charnes, Cooper, and Rhodes (1978) originally proposed the efficiency measurement of the DMUs for constant returns to scale (CRS) where all DMUs are operating at their optimal scale. Later Banker, Charnes, and Cooper (1984) introduced the variable returns to scale (VRS) efficiency measurement model allowing the breakdown of efficiency into technical and scale efficiencies in DEA.

In Figure 1, frontiers determined by economies of scale are presented considering one input and one output for 5 DMUs labeled A through E. The CRS, VRS, and Non-Increasing Returns to Scale (NIRS) frontiers are displayed in the figure. If CRS is assumed then only DMU C would be efficient but DMUs A, C, and E are efficient if VRS is assumed. Where the NIRS and VRS frontiers are equal then decreasing returns to scale exist for those DMUs on the efficient frontier (such as E). Where the two frontiers are unequal, then Increasing Returns to Scale (IRS) exist for those DMUs (such as DMU B). And for the rest of DMUs which are inefficient points can be classified as IRS if the sum of the reference weights is less than unity for the CRS frontier, otherwise as Decreasing Returns to Scale (DRS). The efficiency of observation B is defined as $\theta_{B,input,CRS} = \overline{B_0B_1}/\overline{B_0B}$ for input oriented-constant returns to scale-DEA model and represents that one can obtain the same output by reducing the input by the ratio of $1-\theta_{B,input,CRS}$. And the efficiency for output oriented-constant returns to scale-DEA model is defined as $\theta_{B,output,CRS} = \overline{B_3B}/\overline{B_3C}$ and it represents that one can obtain the same input by increasing the output by the ratio of $1-\theta_{B,output,CRS}$. Accordingly the input oriented efficiency relative to VRS frontier is defined as $\theta_{B,input,VRS} = \overline{B_0B_2}/\overline{B_0B}$. It is noticeable that all efficiency measures of DMU C are the same together regardless of the orientations since the frontiers meet at point C.

It is possible to decompose CRS technical inefficiency into scale efficiency (SE) and "pure" technical efficiency (TE). In Figure 1, $\overline{B_2B}$ contributes to technical efficiency of point B regarding VRS model and $\overline{B_1B}$ contributes to technical efficiency of point B regarding CRS model. Then, $\overline{B_1B_2}$ contributes to scale efficiency.

Figure 2 illustrates the concepts of efficiency, slacks, and references or peers in an intuitive manner using two inputs and one output. The concept of frontier is especially important for the analysis of efficiency, because we measure efficiency as the relative distance to the frontier. For example, firms that are technically inefficient operate at points in the interior of the frontier, while those that are technically efficient operate

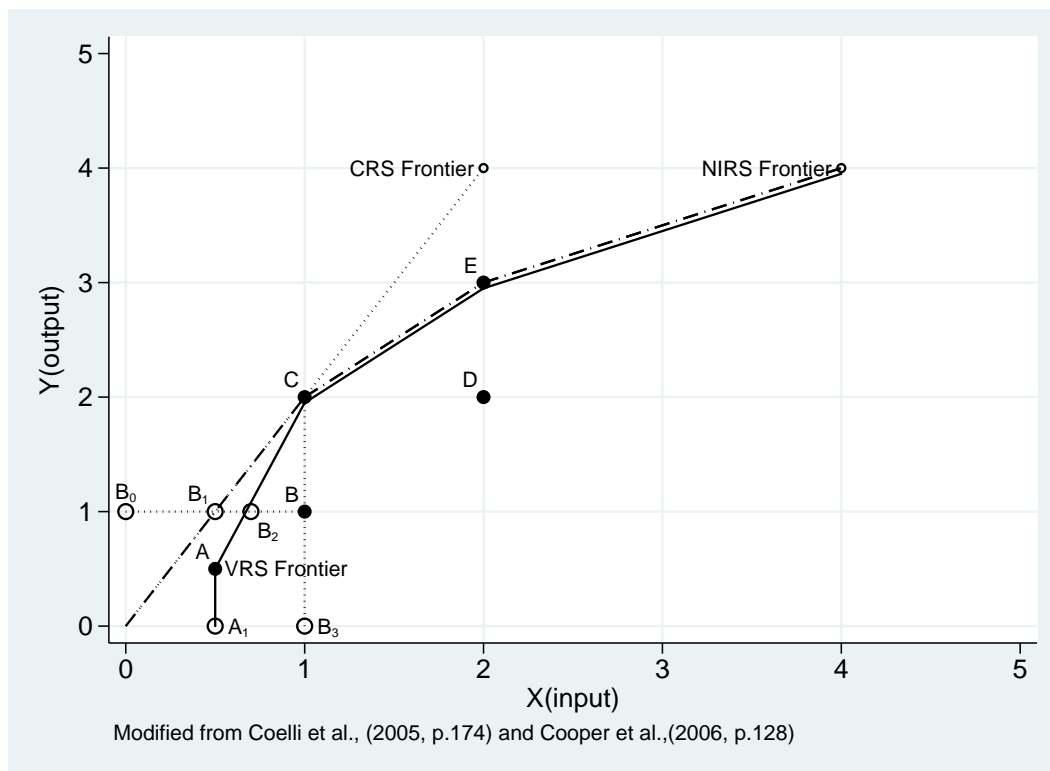


Figure 1: Concepts of Efficiency and Returns to Scale

somewhere along the technology defined by the frontier. The DMU is called efficient when the DEA score equals to one and all slacks are zero(Cooper, Seiford, and Tone, 2006). If only the first condition is satisfied, the DMU is called as efficient in terms of “radial”, “technical”, and “weak” efficiency. If these two conditions are satisfied, the DMU is called efficient in terms of “Pareto-Koopmans” or “strong” efficiency. The technical efficiency of DMUs G and H are defined as $\overline{OG_1}/\overline{OG}$ and $\overline{OH_1}/\overline{OH}$, respectively.

Inefficiency can be seen as how much the inputs must contract along a ray from the origin until it crosses the frontier. For example, for firm G, the measure of technical efficiency is $\overline{OG_1}/\overline{OG}$. Point G_1 is the Farrell efficient point, however, input X_2 could be further reduced and still produce the same output. For this case, firm G has input slack $\overline{CG_1}$. If we disregard the slack and calculate it residually, the DEA model becomes the single-stage DEA model. And the way to reduce the slack and find the Pareto optimal reference set can be further discussed; there are two-stage and multi-stage DEA models available in the literature(Cooper et al., 2006; Coelli et al., 2005). Cherchye and Puyenbroeck (2001) showed that most representative efficient points can be found using a direct approach and may differ from those obtained by multi-stage DEA. The DEA model in this paper provides stage options for single-stage and two-stage which are still

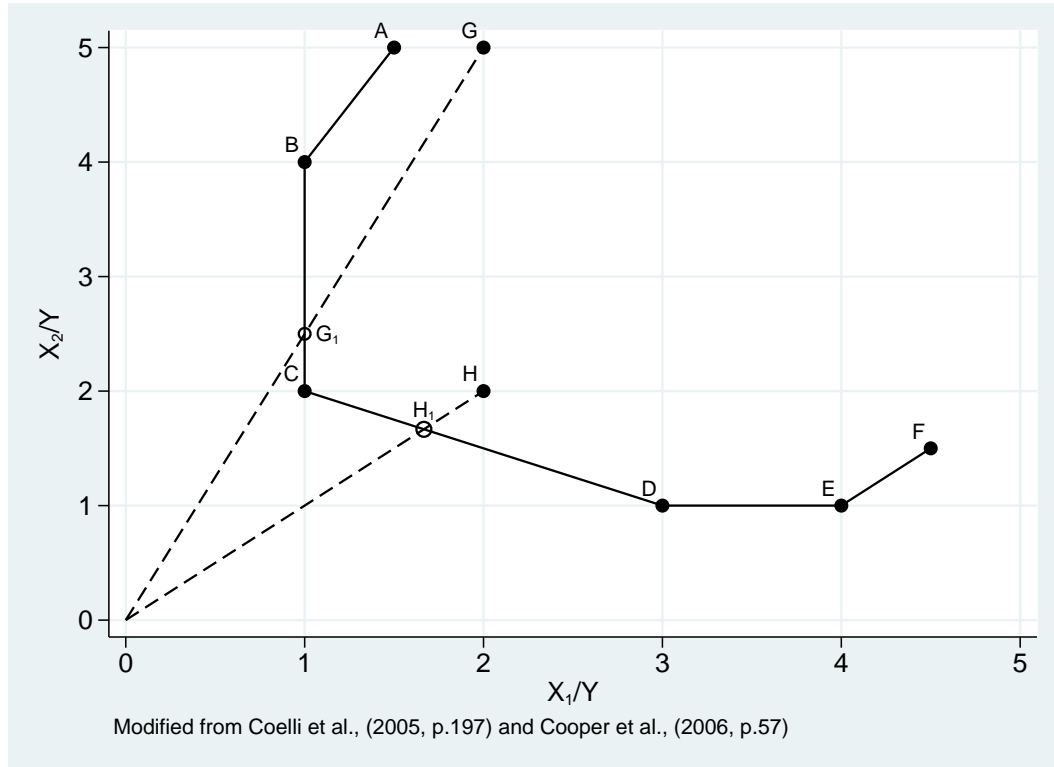


Figure 2: CRS Input Oriented DEA Example

the most prevalent approaches in DEA literature. Reference or peer is a point that an inefficient DMU such as point G targets to move from the Farrell efficient point such as point G_1 to Pareto-Koopman efficient point such as point C in Figure 2 based on the two-stage DEA solution or Pareto optimal solution. However, the slack issues in DEA models disappear as the number of DMUs increases since the DEA piecewise-linear frontier becomes smoother and has less chances to run parallel to the input or output axes.

Free Disposability means that one can produce the same output by wasting resources or increase the output without increasing resources. Strong disposability assumes that it is costless for firms to dispose of inputs or outputs or the isoquant does not bend backwards. In Figure2, the line that links A and F represents for the frontier imposed by weak disposability. The line that links B and E represents the frontier imposed by strong disposability.

Assuming the economic production activities, convexity, strong disposability, and constant returns to scale, we can develop the linear program as a form of piece-wise linear frontier. Input oriented CRS efficiency is defined as Equation (1) by applying

piece-wise linear frontier to input requirement set (Cooper et al., 2006). This enables us to evaluate the efficiency relative to frontier.

$$\max_{\nu, u} z = uy_j \quad (1)$$

subject to $\nu x_j = 1, -\nu X + uY \leq 0, \nu \geq 0, u \geq 0, u_j$ free in sign, where a set of observed DMUs $\{DMU_j; j = 1, \dots, n\}$; x_j, y_j input and output vectors; row vector u, ν output and input multipliers; and X, Y input and output matrices. The goal of input oriented DEA model is to minimize the virtual input, relative to a given virtual output, subject to the constraint that no DMU can operate beyond the production possibility set and the constraint relating to non-negative weights. In practice, the most available DEA programs use the dual forms as expressed in Equation (2) which lower the calculation burden and are virtually the same to Equation (1).

$$\min_{\theta, \lambda} \theta \quad (2)$$

subject to $\theta x_j - X\lambda \geq 0, Y\lambda \geq y_j, \lambda \geq 0$, where λ a semi-positive vector in R^k and θ a real variable. Computational procedure for the Equation (2) can be expressed as the following Equation (3) and Equation (4) problems.

$$\min_{\theta} \theta \quad (3)$$

$$\min_{\lambda, s^+, s^-} \sum -s^+ - s^- \quad (4)$$

subject to $\theta x_j - X\lambda - s^- = 0, Y\lambda + s^+ = y_j, \lambda \geq 0$, where s^+, s^-, λ semi-positive vectors in R^k and θ a real variable. The single-stage DEA model solves Equation (3) and two-stage DEA model solves Equation (3) followed by Equation (4) consecutively. Input oriented CRS model is introduced in this section. However, other variations are easily extendable and available in the most DEA literature including Coelli, Rao, ODonnel, and Battese (2005) and Cooper, Seiford, and Tone (2000, 2006).

3 The dea command

3.1 Syntax

The syntax of the `dea` command is:

```
dea ivars = ovars [if] [in] [using/filename] [, rts(string) ort(string)
stage(#) trace saving(filename)]
```

where `ivars` and `ovars` mean input and output variable lists, respectively.

3.2 Options

`rts(string)` specifies the returns to scale. The default is `rts(crs)`, meaning the constant returns to scale. `rts(vrs)`, `rts(drs)`, and `rts(nirs)` mean the variable returns to

scale, decreasing returns to scale and the nonincreasing returns to scale, respectively.

`ort(string)` specifies the orientation. The default is `ort(i)` or `ort(in)`, meaning the input oriented DEA. `ort(o)` or `ort(out)` means the output oriented DEA.

`stage()` specifies the way to identify all efficiency slacks. The default is `stage(2)`, meaning the two-stage DEA. `stage(1)` means the single-stage DEA.

`trace` lets all the sequences displayed in the result window and also saved in the “`dea.log`” file. The default is to save the final results in the “`dea.log`” file.

`saving(filename)` specifies that the results be saved in `filename.dta`. If the same filename already exists, the existing filename will be replaced by the name of `filename_bak_DMYhms.dta`.

3.3 Description

`dea` requires the user to select the input and output variables from the user designated data file or in the opened data set and solves DEA models by options specified. There are several options to enhance the models. The user can select the desired options according to the particular model that is required.

The `dea` program requires initial data set that contains the input and output variables for observed DMU. Variable names must be identified by `ivars` for input variable and by `ovars` for output variable to allow that `dea` program can identify and handle the multiple input-output data set. And the variable name of DMUs is to be specified by “`dmu`”.

The program has the ability to accommodate unlimited number of inputs/outputs with unlimited number of DMUs. The only limitation is the memory of computer used to run `dea` and the number of observations(DMUs) than the combined number of inputs and outputs to solve the LP problem. The result file reports the information including reference points and slacks in DEA models. These informations can be used to analyze the inefficient DMU, for examples, where the source of inefficiency comes from and how could improve an inefficient unit to the desired level.

`sav(filename)` option creates a `filename.dta` file that contains the results of `dea` including the information of the DMUs, inputs and outputs data used, ranks of Decision Making Units(DMUs), efficiency scores, reference sets, and slacks. The log file `dea.log` will be created in the working directory.

The `dea` program requires the input and output variables and data sets and the options to be defined for the DEA model selection. Based on the data and options specified in the `dea` program, the `dea` program conducts the matrix operations and linear programming to produce the results data sets that are available for print or can be used for further analysis.

3.4 Saved Results

`dea` saves the following in `r()`:

Matrix

`r(dearslt)` $n \times m$ matrix of the results of `dea` command where n is the number of DMUs and m is the information depending on the models specified. Rows correspond to the DMUs and columns correspond to the variables including inputs, outputs, rank(of DMUs scores), theta(efficiency scores), ref.(reference DMUs), input and output slacks, and more depending on the models specified.

4 Applications of dea

4.1 Data

This section provides examples taken using data from Cooper, Seiford, and Tone(2006, p.75, Table 3.7) and Coelli et al.(2005, p.175, Table 6.4) for illustration of the `dea` program. The data of Cooper et al.(2006) consist of five stores that use two inputs, `i_employees`(number of employees as an input variable) and `i_area`(the area of floor as an input variable), to produce two outputs, `o_sales`(the volume of sales as an output variable) and `o_profits`(the volume of profits as an output variable). The data of Coelli et al.(2005) consist of five firms that use one input `i_1` to produce one output `o_1`.

4.2 CRS Input-oriented Two-stage DEA Model

The default of `dea` program specifies constant returns to scale, input-oriented, two-stage DEA Model. If you want to use this specification for your analysis, just use the command `dea` as below using the “`cooper_table3.7.dta`” file. Then you will have the following results. Store E is the only efficient DMU and is referent to all other stores which is equivalent result to Cooper et al.(2006, pp.75-76). The two-stage DEA model provides the information of optimal solution as shown in the result table. For example, the optimal solution of efficiency score(theta), reference weights ($\lambda_A, \lambda_B, \lambda_C, \lambda_D, \lambda_E$), and slack(`i_area`, `i_employee`, `o_sales`, `o_profits`) for Store A are 0.933333, (0, 0, 0, 0, 0.777778), and (11.6667, 0, 0, 0, 0.222222), respectively. Thus the performance of A can be improved by subtracting 11.6667 units from input(area) and 0.777778 unit from output (profits) even after they have reduced all inputs by 6.67% without worsening any other input and output. In other words, because Store A has an efficient score of 93.33%, all inputs (employee and area) could be reduced by 6.67%. In addition, because that store A has an input slack for area of 11.67, 11.67 units of area could be reduced even after they have reduced all inputs by 6.67%.

```
. dea i_employee i_area = o_sales o_profits
```

```
options: RTS(CRS) ORT(IN) STAGE(2)
```

```
ref:
```

```

rank      theta      store_A
dmu:store_A      2      .933333      .
dmu:store_B      3      .888889      .
dmu:store_C      5      .533333      .
dmu:store_D      4      .666667      .
dmu:store_E      1      1      .

ref:      ref:      ref:
store_B   store_C   store_D
dmu:store_A      .      .      .
dmu:store_B      .      .      .
dmu:store_C      .      .      .
dmu:store_D      .      .      .
dmu:store_E      .      .      .

ref:      islack:      islack:
store_E   i_employee   i_area
dmu:store_A      .777778      .      11.6667
dmu:store_B      1.11111      .      3.33333
dmu:store_C      .888889      .      8
dmu:store_D      1.11111      3.33333      .
dmu:store_E      1      .      0

oslack:      oslack:
o_sales      o_profits
dmu:store_A      2.98e-07      .222222
dmu:store_B      7.45e-07      5.88889
dmu:store_C      4.17e-06      2.11111
dmu:store_D      2.98e-06      6.88889
dmu:store_E      .      .

```

4.3 CRS Output-oriented Single-stage DEA Model

If you want to use CRS Output-oriented Single-stage DEA Model, you can specify `dea` program as below using the “cooper.table3.7.dta” file.

```
. dea i_x = o_q,rts(crs) ort(o) stage(1)
```

```

options: RTS(CRS) ORT(OUT) STAGE(1)
CRS-OUTPUT Oriented DEA Efficiency Results:
ref:      ref:      ref:      ref:      ref:
rank      theta      A      B      C      D      E
dmu:A      4      .5      .      .      .333333      .      .
dmu:B      4      .5      .      .      .666667      .      .
dmu:C      1      1      .      .      1      .      .
dmu:D      3      .8      .      .      1.33333      .      .
dmu:E      2      .833333      .      .      1.66667      .      .

islack:      oslack:
i_x      o_q
dmu:A      .      .
dmu:B      .      .
dmu:C      .      .
dmu:D      .      .
dmu:E      .      .

```

The rank of DMUs and efficiency score(theta) as well as residually given reference set(ref:) and slacks(islack: or oslack:) are listed in the above results. Store_C is the only efficient DMU and seems to be referent to all other stores. The result window will display the above result and “dea.log” file that contains the above result will be created in your working directory. The “.” in the result file represents some small numbers less than 10 to the minus 12 powers which mostly can be ignored. However, sometimes when you want to analyze the financial data, the distinction between zero and “.” value may be required to keep the accuracy.

4.4 VRS Input-oriented Single-stage DEA Model

Now the VRS, input-oriented single-stage DEA analysis using the data of coelli_table6.4.dta:

```
. dea i_x = o_q,rts(vrs)stage(1)
```

```
options: RTS(VRS) ORT(IN) STAGE(1)
VRS-INPUT Oriented DEA Efficiency Results:
      rank  theta  ref:  ref:  ref:  ref:  ref:
dmu:A     1     1     A     B     C     D     E
dmu:B     5   .625     .5     .     .5     .     .
dmu:C     1     1     0     .     1     .     .
dmu:D     4     .9     .     .     .5     .     .5
dmu:E     1     1     .     .     0     .     1

      islack:  oslack:
            i_x  o_q
dmu:A      .    0
dmu:B      .    .
dmu:C      .    .
dmu:D      .    .
dmu:E      .    .

VRS Frontier(-1:drs, 0:crs, 1:irs)
      CRS_TE  VRS_TE  NIRS_TE  SCALE  RTS
dmu:A  0.500000  1.000000  0.500000  0.500000  1.000000
dmu:B  0.500000  0.625000  0.500000  0.800000  1.000000
dmu:C  1.000000  1.000000  1.000000  1.000000  0.000000
dmu:D  0.800000  0.900000  0.900000  0.888889 -1.000000
dmu:E  0.833333  1.000000  1.000000  0.833333 -1.000000

VRS Frontier:
```

	dmu	o_q	i_x	CRS_TE	VRS_TE	SCALE	RTS
1.	A	1	2	0.500000	1.000000	0.500000	irs
2.	B	2	4	0.500000	0.625000	0.800000	irs
3.	C	3	3	1.000000	1.000000	1.000000	-
4.	D	4	5	0.800000	0.900000	0.888889	drs
5.	E	5	6	0.833333	1.000000	0.833333	drs

If the returns to scale is set to VRS option, the results show some additional information as shown above. Store D and E are on the decreasing returns to scale(drs) portion of the VRS frontier. On the other hand, Store A and B are on the increasing returns to scale(irs) portion of the VRS frontier.

4.5 VRS Input-oriented Two-stage DEA Model

Now VRS, input oriented, two-stage DEA model can be run using “coelli_6.4.dta”:

```
. dea i_x = o_q,rts(vrs) ort(i)
```

```
options: RTS(VRS) ORT(IN) STAGE(2)
VRS-INPUT Oriented DEA Efficiency Results:
```

	rank	theta	ref: A	ref: B	ref: C	ref: D	ref: E
dmu:A	1	1	1	.	0	.	.
dmu:B	5	.625	.5	.	.5	.	.
dmu:C	1	1	.	.	1	.	.
dmu:D	4	.9	.	.	.5	.	.5
dmu:E	1	1	.	.	0	.	1

```
islack:  oslack:
i_x      o_q
dmu:A    .      .
dmu:B    .      .
dmu:C    .      .
dmu:D    .      .
dmu:E    0      .
```

```
VRS Frontier(-1:drs, 0:crs, 1:irs)
```

	CRS_TE	VRS_TE	NIRS_TE	SCALE	RTS
dmu:A	0.500000	1.000000	0.500000	0.500000	1.000000
dmu:B	0.500000	0.625000	0.500000	0.800000	1.000000
dmu:C	1.000000	1.000000	1.000000	1.000000	0.000000
dmu:D	0.800000	0.900000	0.900000	0.888889	-1.000000
dmu:E	0.833333	1.000000	1.000000	0.833333	-1.000000

```
VRS Frontier:
```

	dmu	o_q	i_x	CRS_TE	VRS_TE	SCALE	RTS
1.	A	1	2	0.500000	1.000000	0.500000	irs
2.	B	2	4	0.500000	0.625000	0.800000	irs
3.	C	3	3	1.000000	1.000000	1.000000	-
4.	D	4	5	0.800000	0.900000	0.888889	drs
5.	E	5	6	0.833333	1.000000	0.833333	drs

Note that the efficiency score(theta) of DMU B is 0.625 and DMUs A and C are the reference DMUs for DMU B. The sum of the reference weights should equal to 1 because VRS option specifies $\sum_{j=1}^n \lambda_j = 1$. It is noted that the sum of the reference weights for DMU B equals to 1 from $(\lambda_A, \lambda_B, \lambda_C, \lambda_D, \lambda_E) = (0.5, 0, 0.5, 0, 0)$. And note that the efficiency scores of all the DMUs are not changed from the single-stage model

shown at the previous section to the current two-stage analysis because there is no slack in this case. This means that the slack level of profits has no effect on the efficiency evaluation. It is noted that Store A, C, and E are the efficient points that inefficient DMUs, B and D, can target to move in input-oriented DEA calculation. Data from Coelli et al.(2005, p.175) are analyzed using saving option:

```
. dea i_x = o.q,rts(vrs) ort(i) sav(coelli_6.4_results)
(output omitted)
```

sav(coelli_6.4_results) option will save the VRS Frontier results, which is shown in section 4.5, in the name of “coelli_6.4.results.dta”. The results matches with the Pareto-Koopman solution of Coelli et al.(2005, p.176) given at Table 6.5 because that slack has no role in this case.

4.6 Second-Stage Regression Analysis using the efficiency scores

The prevalent method to find the determinants of efficiency gaps among DMUs in literature is using the Tobit regression analysis because the efficiency scores are censored for the above the maximum value of efficiency score. The Tobit regression analysis uses the efficiency scores as the dependent variable for the possible candidates of influential variables, see for example Lee, Lee, and Kim (2009).

Now the second stage regression analysis is illustrated using the “seco_stage_regression.dta” data file.

```
. dea i_x = o.q ,sav(seco_stage_regre_results)
(output omitted)
```

```
. use "D:\sjtemp\seco_stage_regre_results",clear
. tobit theta rnd, ul(1)
```

```
Tobit regression                Number of obs   =        20
                               LR chi2(1)           =       46.44
                               Prob > chi2          =        0.0000
Log likelihood = 19.060498      Pseudo R2       =        5.5845
```

theta	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
rnd	.1464713	.0118676	12.34	0.000	.1216322	.1713104
_cons	.3003908	.0360827	8.33	0.000	.2248688	.3759127
/sigma	.0649661	.011565			.0407603	.089172

```
Obs. summary:      0 left-censored observations
                   16 uncensored observations
                   4 right-censored observations at theta>=1
```

The results show that rnd(RD) level is positively related with the CRS efficiency scores of DMUs at the 1 % level of significance.

5 Conclusion

Today many academic researchers recognize Stata as one of the leading packages for data base system and statistical analysis. There are still uncovered areas where managerial organizations are interested in. In particular, optimization procedures in Stata can be further developed to fill in the gaps between parametric and nonparametric analysis. The dea program introduced in this paper is a new application in Stata and is a powerful managerial tool for measuring the efficiency and productivity of DMU.

The dea program application has several advantages including the followings among others:

- It can be used by Stata users with no extra transaction cost to have any DEA software.
- It is flexible to add other DEA models.
- And importantly it provides Stata with managerial tools for reports and statistical analysis as well as optimization procedures.
- The dea program report files can directly feed to other Stata routines for further analysis.

6 Acknowledgments

I thank H. Joseph Newton and an anonymous reviewer for comments.

7 References

- Banker, R. D., A. Charnes, and W. W. Cooper. 1984. Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. *Management Science* 30: 1078–1092.
- Charnes, A., W. W. Cooper, and E. Rhodes. 1978. Measuring the Efficiency of Decision Making Units. *European Journal of Operational Research* 2: 429–444.
- Cherchye, L., and T. V. Puyenbroeck. 2001. A comment on multi-stage DEA methodology. *Operations Research Letters* 28(2): 93–98.
- Coelli, T., D. S. P. Rao, C. J. ODonnel, and G. E. Battese. 2005. *An Introduction to Efficiency and Productivity Analysis, 2nd ed.* 233 Spring Street, New York, NY 10013, USA: Springer.
- Cooper, W., L. M. Seiford, and K. Tone. 2000. *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software.* 233 Spring Street, New York, NY 10013, USA: Springer Science & Business Media Inc
- . 2006. *Introduction to Data Envelopment Analysis and Its Use.* 233 Spring Street, New York, NY 10013, USA: Springer Science & Business Media Inc

Lee, C., J. Lee, and T. Kim. 2009. Defense Acquisition Innovation Policy and Dynamics of Productive Efficiency: A DEA Application to the Korean Defense Industry. *Asian Journal of Technology Innovation* 17(2): 151–171.

About the authors

Yong-bae Ji is a graduate student in the Defense Science and Technology Department at Korea National Defense University in Seoul, Republic of Korea.

Choonjoo Lee (Corresponding author) is an assistant professor in the Defense Science and Technology Department at Korea National Defense University in Seoul, Republic of Korea.